Представление чисел в ЭВМ

Способы представления чисел

- целые положительные числа (без знака)
- целые со знаком
- вещественные нормализованные числа.

Целые числа без знака

<u>Целые числа без знака</u> обычно **занимают в памяти один или два байта** и принимают:

в однобайтовом формате значения от 00000000_2 до 11111111_2 ,

в двубайтовом формате — от $00000000\ 00000000_2$ до $11111111\ 11111111_2$.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$0 \ldots 2^8–1$	0 ... 255
2	$0 \ldots 2^{16}–1$	0 ... 65535

Целые числа со знаком

<u>Целые числа со знаком</u> обычно занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа. Знак “плюс” кодируется **нулем**, а “минус” — **единицей**.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	-2^7 ... $2^7–1$	-128 ... 127
2	-2^{15} ... $2^{15}–1$	-32768 ... 32767
4	$-2^{31}…2^{31}-1$	-2147483648 ... 2147483647

Пример

$72_{10}=1001000_2$

Номера разрядов

а) однобайтовый формат

7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	0

Биты числа

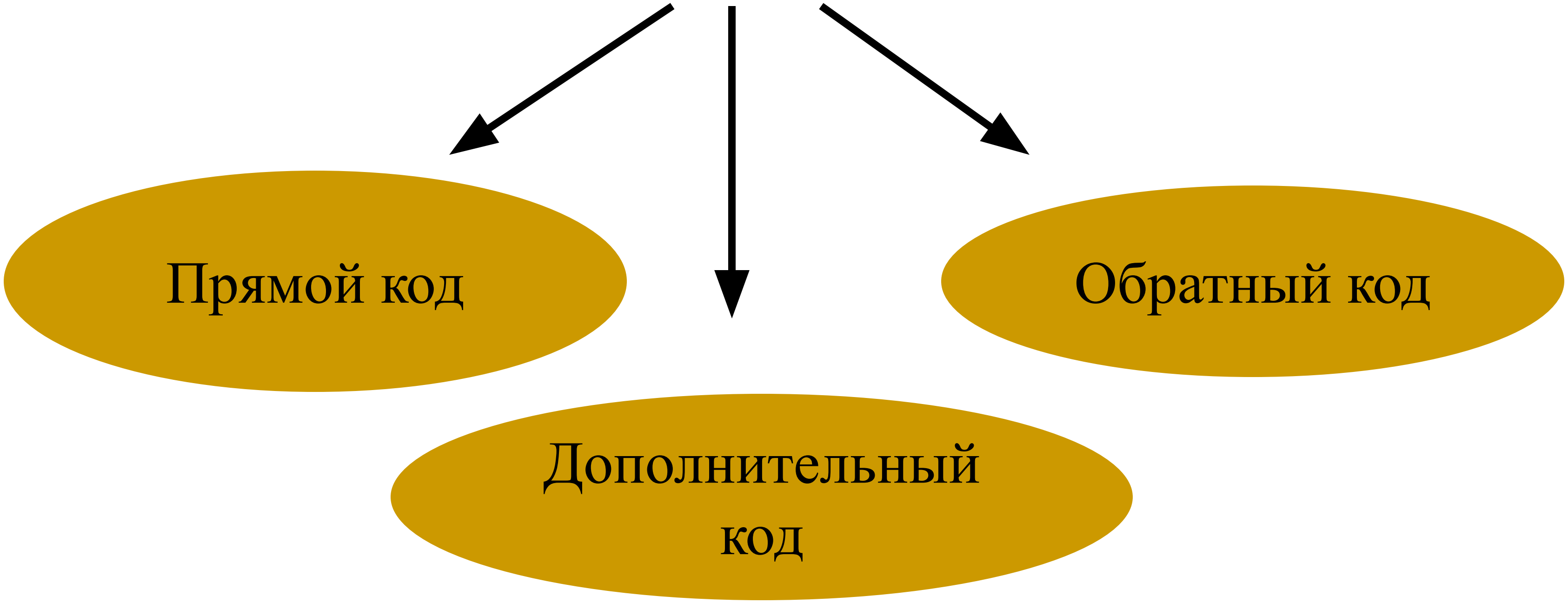
б) двубайтовый формат

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

в) число 65535 в двубайтовом формате

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Целые числа со знаком



Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково - двоичными кодами с цифрой 0 в знаковом разряде

Прямой код

В знаковый разряд помещается цифра знака, а в разряды цифровой части числа — двоичный код его абсолютной величины.

Знак

Прямой код числа: 1

7	6	5	4	3	2	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**

Прямой код числа: -127

7	6	5	4	3	2	1	0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Число

$$\boxed{A_{10} = (-1)^{a_{зн}} \sum_{i=0}^{n-2} a_i 2^i}$$

n-разрядность кода, $a_{зн}$ - значение знакового разряда.

<u>Пример:</u> если разрядность кода равна 4, то

$$1101 = (-1)^1[1x2^0+0x2^1+1x2^2]=-5$$

Обратный код

Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями.

Пример:

число: -1, модуль 0 0000001, обратный код 1 1111110

число: -127, модуль 0 1111111 , обратный код 1 0000000

$$\boxed{A_{10} = a_{зн}(-2^{n-1}+1) + \sum_{i=0}^{n-2} a_i 2^i}$$

n-разрядность машинного слова, $a_{зн}$=0 для положительных чисел, $a_{зн}$=1 для отрицательных чисел.

$$1010 = 1*(-2^3+1)+[0x2^0+1x2^1+0x2^2] = -7+2=-5$$

Дополнительный код

Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду

число: -1 = обратный код 1 1111110
число: -127 = обратный код 1 0000000

Дополнительный код числа: -1

7	6	5	4	3	2	1	0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Дополнительный код числа: -127

7	6	5	4	3	2	1	0
1	**0**	**0**	**0**	**0**	**0**	**0**	**1**

Обычно *отрицательные* десятичные числа при вводе в машину *автоматически* преобразуются в *обратный* или *дополнительный* двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит *обратное преобразование* в отрицательные десятичные числа.

Дополнительный код

алгоритм перевода отрицательных чисел в положительные

I вариант. Переписать исходную последовательность битов числа справа налево до первой единицы, включая ее. Остальные биты инвертировать.	$6_{10}=0110_2$ $-6_{10}=\underline{1010}_2$
II вариант. Дополнительный код=логическое дополнение (все биты инвертированы)+1	$-6_{10}=1001_2+1$ $=\underline{1010}_2$

Число + его дополнительный код =0

Дополнительный код

Представление в двоичном дополнительном коде в случае 3-битного кодирования чисел:

Набор битов	Значение
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

Дополнительный код

Для дополнительного кода справедливо следующее соотношение:

$$A_{10} = a_{зн}(-2^{n-1}) + \sum_{i=0}^{n-2} a_i 2^i$$

где n-разрядность машинного слова, $a_{зн}$=0 для положительных чисел, $a_{зн}$ =1 для отрицательных чисел.

Пример: 1101 = 1*(-2^3)+[1x2^0+1x2^1+0x2^2]=-8+3=-5

Операции над целыми числами

- Сложение.
- Вычитание. В большинстве случаев операция вычитания не используется, вместо нее производится сложение обратных или дополнительных кодов уменьшаемого и вычитаемого.
- Умножение
- Целочисленное деление и нахождение остатка от деления

Сложение обратных кодов

А и В положительные

Десятичная запись	Двоичные коды
3	0 0000011
+ 7	+ 0 0000111
10	0 0001010

Сложение обратных кодов

А положительное, **В** отрицательное и по абсолютной величине больше, чем **А**

Десятичная запись	**Двоичные коды**	
3	**0 0000011**	
+ -10	**+ 1 1110101**	**Обратный код -10**
-7	**1 1111000**	**Обратный код -7**

Сложение обратных кодов

А положительное, **В** отрицательное и по абсолютной величине меньше, чем **А**

Десятичная запись	Двоичные коды	
10	0 0001010	
+ -3	+ 1 1111100	Обратный код -3
7	0 0000110	
	+ 1	
	0 0000111	

Сложение обратных кодов

А и **В** отрицательные

Десятичная запись	**Двоичные коды**	
-3	1 1111100	**Обратный код -3**
+ -7	+ 1 1111000	**Обратный код -7**
-10	1 1110100	
	+ 1	
	1 1110101	**Обратный код -10**

Сложение обратных кодов

При сложении может возникнуть ситуация, когда старшие разряды результата операции не помещаются в отведенной для него области памяти.

Такая ситуация называется **переполнением разрядной сетки** формата числа.

Случай переполнения возможен и для обратных и для дополнительных кодов.

Сложение дополнительных кодов

А и В положительные

Десятичная запись	Двоичные коды
3	$0\ 0000011$
$+\ 7$	$+\ 0\ 0000111$
10	$0\ 0001010$

Сложение дополнительных кодов

А положительное, **В** отрицательное и по абсолютной величине больше, чем **А**

Десятичная запись	**Двоичные коды**	
3	**0 0000011**	
+ -10	**+ 1 1110110**	**Дополнительный код -10**
-7	**1 1111001**	**Дополнительный код -7**

Сложение дополнительных кодов

А положительное, **В** отрицательное и по абсолютной величине меньше, чем **А**

Десятичная запись	**Двоичные коды**	
10	0 0001010	
+ -3	+ 1 1111101	**Дополнительный код -3**
7	1 0000111	
0		

Перенос отбрасывается

Сложение дополнительных кодов

А и **В** отрицательные

Десятичная запись	**Двоичные коды**	
-3	1 1111101	**Дополнительный код -3**
+ -7	+ 1 1111001	**Дополнительный код -7**
-10	1 1 1110110	**Дополнительный код -10**

Перенос отбрасывается

Формы представления чисел

- С фиксированной точкой
- С плавающей точкой

Преобразование чисел из естественной формы в нормализованную

- <u>Число больше 1.</u>

 Перемещение разделителя по числу влево до тех пор, пока не исчезнет целая часть. Нормализация влево. $N_{\leftarrow}$

$$\mathbf{N_{\leftarrow}[1234,56]=0.123456*10^{4}}$$

$$\mathbf{N_{\leftarrow}[23,4*10^{6}]=0.234*10^{7}}$$

- <u>Число меньше 1.</u>

 Перемещение разделителя по числу вправо до тех пор, пока первая цифра после разделителя не станет ненулевой. Нормализация вправо. $N_{\rightarrow}$

$$\mathbf{N_{\rightarrow}[0.0003]=0.3*10^{-3}}$$

Вещественные числа

Любое число *N* в системе счисления с основанием *q* можно записать в виде $N = M * q^p$, где *M* называется мантиссой числа, а *p* — порядком. Такой способ записи чисел называется представлением с плавающей точкой.

Для удобства отображения чисел, принимающих значения из достаточно широкого диапазона, используется форма записи чисел с **порядком основания системы счисления**.

Например:

$$1.25*10^0 = 0.125*10^1 = 0.0125*10^2 = \ldots ,$$

или:

$$12.5*10^{-1} = 125.0*10^{-2} = 1250.0*10^{-3} = \ldots .$$

Вещественные числа

Мантисса должна быть **правильной дробью**, первая цифра которой отлична от нуля: **M** из [0.1, 1).

Такое, наиболее выгодное для компьютера, представление вещественных чисел называется **нормализованным**.

Мантиссу и порядок q-ичного числа принято записывать в системе с основанием q, а само основание — в десятичной системе.

Примеры нормализованного представления вещественного числа

Десятичная система

$753.15 = 0.75315*10^{3}$ $-0.000034 = -0.34*10^{-4}$

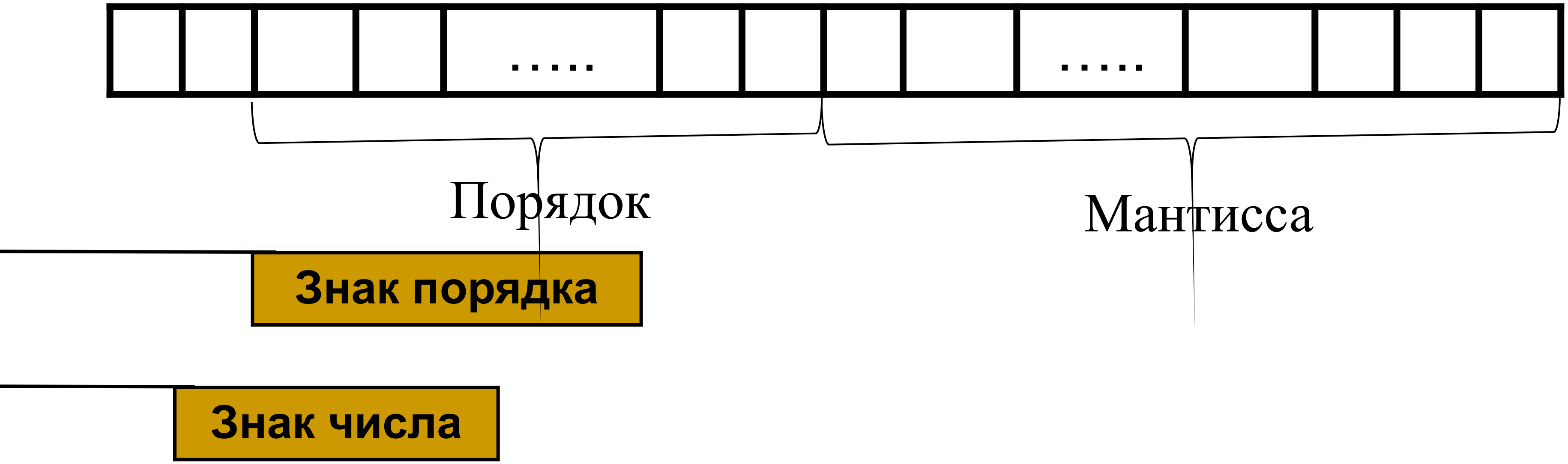
Двоичная система

$-101.01 = -0.10101*2^{11}$ (порядок $11_2 = 3_{10}$)

$-0.000011 = 0.11*2^{-100}$ (порядок $-100_2 = -4_{10}$)

Формат представления вещественных чисел

При хранении числа с плавающей точкой отводятся **разряды для мантиссы, порядка, знака числа и знака порядка:**



1. Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа.
2. Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от нуля числа до наибольшего числа, представимого в машине при заданном формате.

Пример записи чисел в нормализованном виде в четырехбайтовом формате с семью разрядами для записи порядка.

Число $6.25_{10} = 110.01_2 = +0.11001 \cdot 2^{+11}$

31	30	…							22									2	1	0
0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	…..	0	0	0

Порядок

Мантисса

Знак порядка

Знак числа

Пример записи чисел в нормализованном виде в четырехбайтовом формате с семью разрядами для записи порядка.

2. Число $-0.125_{10} = -0.001_2 = -0.1 * 2^{-10}$

(отрицательный порядок записывается в дополнительном коде)

31 30 22 2 1 0

1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	…..	0	0	0

Порядок

Мантисса

Знак порядка

Знак числа

Характеристики форматов вещественных чисел

Форматы вещественных чисел	Размер в байтах	Примерный диапазон абсолютных значений	Количество значащих десятичных цифр
Одинарный	4	$10^{-45} \ldots 10^{38}$	7 или 8
Вещественный	6	$10^{-39} \ldots 10^{38}$	11 или 12
Двойной	8	$10^{-324} \ldots 10^{308}$	15 или 16
Расширенный	10	$10^{-4932} \ldots 10^{4932}$	19 или 20

Форма представления чисел с плавающей точкой позволяет записывать числа с высокой точностью и из весьма широкого диапазона.

Арифметические операции с вещественными числами

1. Сложение .

$$X_1 = M_1 \cdot 10^{k_1} \qquad X_2 = M_2 \cdot 10^{k_2}$$

a) $\Delta k = |k_1 - k_2|$

b) если $k1 > k2$, то $M = M_1 + M_2 10^{-\Delta k}$ $k = k_1$

иначе $M = M_2 + M_1 10^{-\Delta k}$ $k = k_2$

c) если $10^{-1} <= M < 1$, то вывод результата в виде $M \times 10^k$, иначе предварительная нормализация

Арифметические операции с вещественными числами

1. <u>Сложение . Пример.</u>

Сложить двоичные нормализованные числа $0.10111 \,.\, 2^{-1}$ и $0.11011 \,.\, 2^{10}$. Разность порядков слагаемых здесь равна трем, поэтому перед сложением мантисса первого числа сдвигается на три разряда вправо:

$$
\begin{array}{rl}
+ & 0.00010111 * 2^{10} \\
 & 0.11011 \quad\;\; * 2^{10} \\
\hline
 & 0.11101111 * 2^{10}
\end{array}
$$

Арифметические операции с вещественными числами

Пример.

X1=0.87654 * 10^1, X2=0.94567*10^2. Пусть под запись мантиссы отводится 5 разрядов.

1. Δk=1, $k_1<k_2$ следовательно $k_1=k_2=2$ (уравняли порядки)
2. мантиссу числа X1 сдвигаем на один разряд влево (пропадет 4)
3. новая мантисса равна 0,94567+0,08765=1,03332
4. мантисса вышла за допустимый интервал >1.
5. нормализуя, получим мантиссу 0,10333 (теряем 2) и порядок увеличиваем на 1.

Ответ: X=0,10333*10^3 = 103,3324.

Арифметические операции с вещественными числами

1. Вычитание сводится к сложению с дополнительным кодом.
2. Умножение производится по правилу – мантиссы перемножаются, а порядки складываются. Если нужно, то полученное число нормализуется.
   Пример
   $(0.11101 * 2^{101}) * (0.1001 * 2^{11}) = (0.11101 * 0.1001) * 2^{(101+11)} = 0.100000101 * 2^{1000}$.
3. Деление производится по правилу – мантиссы делятся (делимое на делитель), а порядки вычитаются (порядок делителя из порядка делимого). Если нужно, то полученное число нормализуется
   $0.1111 * 2^{100} : 0.101 * 2^{11} = (0.1111 : 0.101) * 2^{(100-11)} = 1.1 * 2^{1} = 0.11 * 2^{10}$.

Вопросы?